Programação Shell Linux

Conceitos Básicos – Parte II

Objetivos

- Apresentar o uso de caracteres especiais;
- Conhecer o redirecionamento de caracteres;

Uso de Aspas, Apóstrofos e Barra Invertida

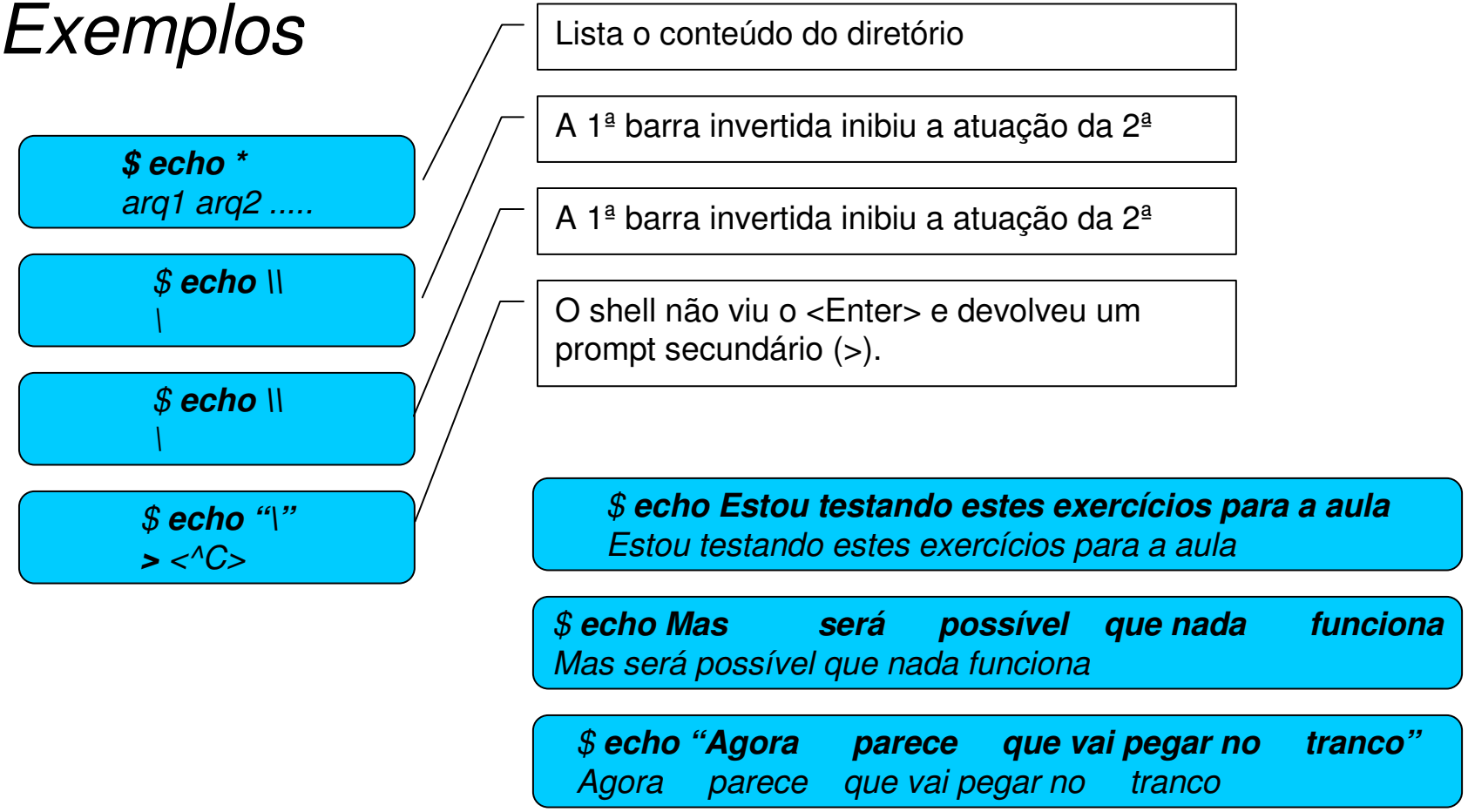
- ***Aspas (“)***
  - *Para usar um caractere sem que o **shell** tente interpretar o seu significado;*
  - *O **shell** **ignora** o significado do caractere entre elas;*
  - *Isto não ocorre para o caso de um $ (cifrão), ` (crase) ou \ (barra invertida).*

Uso de Aspas, Apóstrofos e Barra Invertida

- ***Apóstrofos (')***
  - *Mais restritivos;*
  - *Todos os caracteres são ignorados;*
- ***Barra Invertida (\)***
  - *O shell **ignora um e somente um** caractere que segue a barra invertida;*
  - *É interpretado como um aviso de continuação da linha, devolvendo um prompt secundário **(>)**;*
  - *O shell não enxerga o <enter>, que segue a contrabarra.*

Uso de Aspas, Apóstrofos e Barra Invertida

- *Exemplos*



Crase e Parênteses

- ***Crases (`)***
  - *Usadas para avisarmos ao Shell que o que está entre elas é um **comando** e para darmos **prioridade** em sua execução.*

$ echo "O nome deste computador é `uname –n`"
O nome deste computador é localhost.localdomain

$ echo "O nome deste computador é uname –n"
O nome deste computador é uname -n

a **Apresenta o que está entre aspas e executa o que está entre crases**

b **Apresenta o que está entre aspas**

Crase e Parênteses

$ (cd ../exercicios; ls)
Arq1
Arq2
Arq3

$ pwd
/home/aluno

$ echo " O nome deste computador é $(uname –n)"
O nome deste computador é manguinhos

a **Os parênteses invocaram um novo *shell* que foi para o segundo diretório, listou o seu conteúdo, e morreu....**

b **O *shell* mantém-se no mesmo diretório**

c **Utilizada no lugar da crase, porém, nem todas as distribuições a implementaram.**

Direcionamento e Redirecionamento

- ***A maioria dos comandos têm uma entrada, uma saída e pode gerar erros;***
- ***Entrada Padrão***
  - ***Teclado (stdin).***
- ***Saída Padrão***
  - ***Terminal (stdout).***
- ***Erro Padrão***
  - ***Terminal (stderr).***

Direcionamento e Redirecionamento

Redirecionamento de Saída

> Redireciona a saída de um comando para um arquivo especificado, inicializando-o, caso não exista ou destruindo o seu conteúdo anterior

>> Redireciona a saída de um comando para um arquivo especificado, anexando-o ao seu fim. Caso este arquivo não exista, será criado.

2> Redireciona os erros gerados por um comando para o arquivo especificado. Mesmo que não ocorra erro na execução do comando, o a arquivo será criado.

Direcionamento e Redirecionamento

Redirecionamento de Entrada

< Avisa ao Shell que a entrada padrão não será o teclado, mas sim o arquivo especificado.

<< Serve para indicar ao Shell que o escopo de um comando começa na linha seguinte e termina quando encontra uma linha cujo conteúdo seja unicamente o *label* que segue o sinal.

Direcionamento e Redirecionamento

Redirecionamentos Especiais

| Este é o famoso *pipe*, e serve para direcionar a saída de um comando para a entrada de outro.

tee Captura a saída de um comando com pipe, copiando o que está entrando no tee para a saída padrão e outro comando ou arquivo.

Direcionamento e Redirecionamento

Exemplo 1

```
$ ftp –ivn remocomp << FimFTP >> //tmp/  2>> /tmp/$$
>     user fulano senha
>     binary
>     get arqnada
FimFTP

$
```

a. Fizemos um FTP para remocomp

b. << FimFTP avisa ao shell que, até que o label FimFTP seja encontrado em alguma linha, todas as linhas intermediárias pertencem ao comando FTP e não deve ser interpretado pelo shell.

c. >> /tmp/$$ significa que as mensagens do ftp deverão ser anexadas ao arquivo /tmp/<num. do processo>

d. 2>> /tmp/$$ significa que as mensagens de erro do FTP deverão ser anexadas ao arquivo /tmp/<num. do processo>.

A variável $$ deve ser usada para compor nomes de arquivos temporários, Evita conflitos de permissões de diferentes usuários sobre o *referido* arquivo

Direcionamento e Redirecionamento

Exemplo 2

```
$ mail fulano << FimMail
Sr. Fulano,
Vitória, `date`
O nosso candidato à Presidente tem um dossiêzinho
com os seguintes arquivos:
`ls –l`

Atenciosamente, Psicólogo
FimMail
```

a. Enviamos um email para Fulano

b. O conteúdo termina com o label FimMail.

c. O shell identificou os comandos *date* e *ls* entre crases e resolveu-os.

Cuidado para não colocar espaços em branco antes ou após um *label*. Erro Muito comum em *Scripts*. É um erro de difícil detecção.

Direcionamento e Redirecionamento

Exemplo 3

```
$ rm arquivo1 2> /dev/null
```

a) Caso o arquivo não exista, a mensagem de erro não será exibida na tela

b) A mensagem será enviada para */dev/null*

Exemplo 4

```
$ echo Atualmente existem `who | wc –l` usuários conectados
```

a) Apresenta a mensagem após o comando *echo*

b) *who* - Conta quantos usuários estão conectados, e lista (*wc*) o número de linhas.

DIVERSÃO! *EXERCÍCIOS.*